Table of contents

```
                                        ;  Finished
                                        ;
001546  012600                  9$:     MOV     (SP)+,R0        ;Get message length
001550  012604                          MOV     (SP)+,R4
001552  012603                          MOV     (SP)+,R3
001554  012602                          MOV     (SP)+,R2
001556  012601                          MOV     (SP)+,R1
001560  000207                          RETURN
```

```
                                        .TITLE  TSMSG -- TSX-Plus Message Transfer System
                                        .ENABL  LC
                                        .DSABL  GBL
                                        .ENABL  AMA
                                ;
                                ;  Copyright (c) 1977,1978,1979,1980,1981,1982,1983,1984.
                                ;
                                ;  S&H Computer Systems, Inc.
                                ;  Nashville, Tennessee
                                ;
                                ;  The routines in this module implement the TSX-Plus interjob
                                ;  message communication system.
                                ;
000000                                  .CSECT  TSMSG
000000  052077                  TSMSG:  .RAD50  /MSG/            ;System overlay ID
                                ;
                                ;  Global definitions
                                ;
                                        .GLOBL  MSEND,MSGCK,MSGWT,TSMSG
                                        .GLOBL  MSGINI,MSGTOP,MSGABT,MBMHD
                                ;
                                ;  Global references
                                ;
                                        .GLOBL  MB$BUF,MB$FLK,EMTBLK,VMAXMC,EB$BUF,EB$RTN
                                        .GLOBL  CORUSR,MR$LNK,MR$JOB,MB$FLK,MR$RTN,QF$SCR
                                        .GLOBL  MB$NAM,PUTUCH,S$HICP,EB$NAM,GETUCH,S$IOFN
                                        .GLOBL  MB$REQ,MU$SIZ,QCOMPL,LITIME,GETQ,CQ$PRI
                                        .GLOBL  MB$$SZ,CHKABT,INTERR,QHDSPN,EB$SIZ,CQ$RO
                                        .GLOBL  MU$TXT,VMSCHR,QHIPRI,URO,CQ$RTN,VPAR6,CQ$FLG
                                        .GLOBL  LPRI,VMXMRB,LPRI,S$MSWT,MR$$SZ,MR$BUF,VMXMSG
                                        .GLOBL  MR$UBA,MR$URS,LSTATE,CQ$RNS,MU$FLK,LCMQHD,CQ$JOB
                                        .GLOBL  KPAR5,CQ$PA5,MU$JOB,CQ$R1,CQ$CP,CP$STD
                                ;
                                ;  Symbolic equates for EMT error codes
                                ;
000001                  ERR01   =       1               ;All message channels are busy
000002                  ERR02   =       2               ;No free message queue entries
000003                  ERR03   =       3               ;No message was queued on specified channel
000004                  ERR04   =       4               ;Message was longer than specified buffer
000005                  ERR05   =       5               ;No free message request blocks
                                ;
                                ;  Data areas
                                ;
000002  000000          MBMHD:  .WORD   0               ;List head of active message channel blocks
000004  000000          MCBFHD: .WORD   0               ;List of free message channel blocks
000006  000000          MRBFHD: .WORD   0               ;List of free message request blocks
000010  000000          MTBFHD: .WORD   0               ;List of free message text blocks
000012                  CHNNAM: .BLKW   3               ;Used to hold channel name string
```

```
                                        .SBTTL  MSGINI -- Message system initialization
                                ;-----------------------------------------------------------------------
                                ;  MSGINI is called during system initialization to initialize the
                                ;  data areas used for message communication.
                                ;  The message tables are allocated within this system overlay
                                ;  starting just beyond the end of the code.
                                ;
000020  010246                 MSGINI: MOV     R2,-(SP)
000022  010346                         MOV     R3,-(SP)
000024  010446                         MOV     R4,-(SP)
000026  010546                         MOV     R5,-(SP)
                                ;
                                ;  Start data areas above top of code
                                ;
000030  012702  002412'                 MOV     #MSGTOP,R2      ;Get address above top of code
                                ;
                                ;  Allocate message channel blocks
                                ;
000034  010237  000004'                 MOV     R2,MCBFHD       ;Set pointer to 1st free message channel blk
000040  013700  0000000                 MOV     VMAXMC,R0       ;Get # blocks to allocate
000044  010203                  1$:     MOV     R2,R3           ;Get address of current block
000046  062703  0000000                 ADD     #MB$$SZ,R3      ;Get address of next block
000052  010362  0000000                 MOV     R3,MB$FLK(R2)   ;Set pointer to next block
000056  010302                          MOV     R3,R2           ;Get address of next block
000060  077007                          SOB     R0,1$           ;Loop if need to allocate more
000062  005062  000000C                 CLR     MB$FLK-MB$$SZ(R2);Clear link in last block
                                ;
                                ;  Allocate message request blocks
                                ;
000066  013700  0000000                 MOV     VMXMRB,R0       ;Get # to allocate
000072  001413                          BEQ     3$              ;Br if none needed
000074  010237  000006'                 MOV     R2,MRBFHD       ;Set pointer to 1st free block
000100  010203                  2$:     MOV     R2,R3           ;Get address of current block
000102  062703  0000000                 ADD     #MR$$SZ,R3      ;Get address of next block
000106  010362  0000000                 MOV     R3,MR$LNK(R2)   ;Set pointer to next block
000112  010302                          MOV     R3,R2           ;Get address of next block
000114  077007                          SOB     R0,2$           ;Loop if need to allocate more
000116  005062  000000C                 CLR     MR$LNK-MR$$SZ(R2);Clear link in last block
                                ;
                                ;  Allocate message text blocks
                                ;
000122  013700  0000000         3$:     MOV     VMXMSG,R0       ;Get # buffers to allocate
000126  001433                          BEQ     9$              ;Br if none wanted
000130  005037  0000000                 CLR     VMXMSG          ;Count actual buffers allocated
000134  010237  000010'                 MOV     R2,MTBFHD       ;Set pointer to 1st free block
000140  013704  0000000                 MOV     VMSCHR,R4       ;Get space needed for message text
000144  062704  0000010                 ADD     #<MU$TXT+1>,R4  ;Add space for message header
000150  042704  000001                  BIC     #1,R4           ;Make sure size is even
000154  012705  0000000                 MOV     #VPAR6,R5       ;Get highest legal address for data areas
000160  160405                          SUB     R4,R5           ;Reduce by size of a block
000162  160405                          SUB     R4,R5
000164  010203                  4$:     MOV     R2,R3           ;Get address of current block
000166  060403                          ADD     R4,R3           ;Get address of next block
000170  020305                          CMP     R3,R5           ;Would this block overflow?
000172  103006                          BHIS    6$              ;Br if yes
000174  005237  0000000                 INC     VMXMSG          ;Count another actual block
000200  010362  0000000                 MOV     R3,MU$FLK(R2)   ;Set link to next block
```

```
     58 000204  010302                          MOV     R3,R2           ;Get address of next block
     59 000206  077012                          SOB     R0,4$           ;Loop if more to allocate
     60 000210  160403                  6$:     SUB     R4,R3           ;Get address of last block
     61 000212  005063  0000000                 CLR     MU$FLK(R3)      ;Say it is at end of list
     62                                 ;
     63                                 ;  Finished
     64                                 ;
     65 000216  012605                  9$:     MOV     (SP)+,R5
     66 000220  012604                          MOV     (SP)+,R4
     67 000222  012603                          MOV     (SP)+,R3
     68 000224  012602                          MOV     (SP)+,R2
     69 000226  000207                          RETURN
```

```
                                        .SBTTL  MSEND  -- Queue a message
                                ;-----------------------------------------------------------------------
                                ;  The message send EMT is used to queue a message for another job.
                                ;
                                ;  Inputs:
                                ;   R0 = Address of EMT argument block.
                                ;
000230                          MSEND:
                                ;
                                ;  See if there is an existing message channel block for the specified
                                ;  message channel.
                                ;
000230  004737  001562'               CALL    MBSRCH          ;Search for existing message channel block
000234  103007                       BCC     1$              ;Br if found existing message channel block
                                ;
                                ;  There is no existing message channel block for this message channel.
                                ;  Create a new message channel block.
                                ;
000236  004737  002152'               CALL    GETMCB          ;Get a new message channel block
000242  103004                       BCC     1$              ;Br if got a message channel block
000244  112737  000001  0000000      MOVB    #ERR01,INTERR   ;Return error 1
000252  000453                       BR      20$
                                ;
                                ;  At this point, R2 contains the address of the message channel block.
                                ;  Create a message text block.
                                ;
000254  004737  002242'         1$:   CALL    GETMTB          ;Get a message text block
000260  103006                       BCC     3$              ;Br if got one
000262  112737  000002  0000000      MOVB    #ERR02,INTERR   ;Return error # 2
000270  004737  002272'               CALL    FREMCB          ;Free the MCB we got if it is idle
000274  000442                       BR      20$
                                ;
                                ;  At this point,
                                ;   R2 = Address of message control block.
                                ;   R4 = Address of message text block
                                ;  Link the new message text block onto the list for the message channel.
                                ;
000276  010205                  3$:   MOV     R2,R5           ;Get address of message control block
000300  062705  0000000              ADD     #MB$BUF,R5      ;Point to cell with addr of 1st text block
000304  005715                  4$:   TST     (R5)            ;Is there another text block on chain?
000306  001404                       BEQ     5$              ;Br if reached end of chain
000310  011505                       MOV     (R5),R5         ;Chain forward to next text block
000312  062705  0000000              ADD     #MU$FLK,R5      ;Point to cell with forward link in text blk
000316  000772                       BR      4$              ;Continue following chain
000320  010415                  5$:   MOV     R4,(R5)         ;Add new message text block to end of chain
                                ;
                                ;  Save job index number of job sending the message in message text block
                                ;
000322  113764  0000000 0000000      MOVB    CORUSR,MU$JOB(R4);Save index # of job sending message
                                ;
                                ;  Move the message text from the user's buffer into the message text buffer
                                ;
000330  013705  0000000              MOV     EMTBLK+EB$SIZ,R5;Get size of user's message
000334  020537  0000000              CMP     R5,VMSCHR       ;Is it longer than buffer space?
000340  101405                       BLOS    6$              ;Br if length ok
000342  112737  000004  0000000      MOVB    #ERR04,INTERR   ;Return error code 4 if message is too long
000350  013705  0000000              MOV     VMSCHR,R5       ;Truncate message length
```

```
     58 000354  010564  000000G         6$:     MOV     R5,MU$SIZ(R4)   ;Set message size in message text block
     59 000360  001406                          BEQ     7$              ;Br if message length is zero
     60 000362  062704  000000G                 ADD     #MU$TXT,R4      ;Point to start of message text buffer
     61 000366  013703  000000C                 MOV     EMTBLK+EB$BUF,R3;Get address of user's message buffer
     62 000372  004737  001346'                 CALL    MSGGET          ;Get message into buffer
     63                                     ;
     64                                     ;  We have created a message control block and a messate text block
     65                                     ;  and added the message text block to the end of the list of pending
     66                                     ;  messages for the message channel.
     67                                     ;  Now call routine to see if any users are waiting for a message on
     68                                     ;  this message channel.
     69                                     ;
     70 000376  004737  000672'         7$:     CALL    MSGACT          ;Try to activate any waiting users
     71                                     ;
     72                                     ;  Finished
     73                                     ;
     74 000402  000207                  20$:    RETURN
```

```
                                        .SBTTL  MSGCK  -- Check for pending message
                                ;-----------------------------------------------------------------------
                                ;  The message check EMT is used to determine if a message is pending
                                ;  on a message channel.  If there is a message, the message is passed
                                ;  to the user.  If there is no message, an error return occurs.
                                ;
000404                          MSGCK:
                                ;
                                ;  See if there is an existing message channel block
                                ;
000404  004737  001562'                 CALL    MBSRCH          ;Search for existing message channel block
000410  103403                          BCS     1$              ;Br if no messages
                                ;
                                ;  There is an existing message channel block.
                                ;  See if there are any pending messages for this channel.
                                ;
000412  005762  0000000                 TST     MB$BUF(R2)      ;Are there any pending messages?
000416  001006                          BNE     MSGWT           ;Br if yes -- Go move 1st to user
                                ;
                                ;  There are no pending messages.
                                ;  Return error code 3.
                                ;
000420  112737  000003  0000000 1$:     MOVB    #ERR03,INTERR   ;Return error code 3
000426  005037  0000000                 CLR     URO             ;Say message length = 0
000432  000207                          RETURN
```

```
                                        .SBTTL  MSGWT  -- Wait for a message
                                ;-------------------------------------------------------------
                                ;  The message wait EMT attempts to get a pending message and if none
                                ;  is available suspends the job until one becomes available.
                                ;
000434                          MSGWT:
                                ;
                                ;  See if this is a request to get a message with wait or with a
                                ;  completion routine.
                                ;
000434  105737  0000000                 TSTB    EMTBLK          ;If channel # non-zero then compl routine
000440  001063                         BNE     MSRCPL          ;Read with completion
                                ;
                                ;  See if there is an existing message channel block for the specified
                                ;  message channel.
                                ;
000442  004737  001562'                 CALL    MBSRCH          ;Search for existing message channel block
000446  103007                         BCC     1$              ;Br if found one
                                ;
                                ;  There is no existing message channel block.
                                ;  Create a new one.
                                ;
000450  004737  002152'                 CALL    GETMCB          ;Get a new message channel block
000454  103004                         BCC     1$              ;Br if we got a message control block
000456  112737  000001  0000000       MOVB    #ERRO1,INTERR   ;Return error code 1
000464  000450                         BR      20$
                                ;
                                ;  Create a message request block for this job
                                ;
000466  004737  001710'         1$:    CALL    ADDMRB          ;Add message request block
000472  103006                         BCC     6$              ;Br if we got a message request block
                                ;
                                ;  Error, we could not get a free message request block
                                ;
000474  112737  000005  0000000       MOVB    #ERRO5,INTERR   ;Return error code 5
000502  004737  002272'                 CALL    FREMCB          ;See if we need to free the message control bk
000506  000437                         BR      20$
                                ;
                                ;  At this point we have created a message request block and added it to
                                ;  the list of pending request blocks for this channel.
                                ;  Call MSGACT to see if the request can be satisfied immediately.
                                ;   R2 = Address of message channel block.
                                ;   R4 = Address of message request block.
                                ;
000510  004737  000672'         6$:    CALL    MSGACT          ;Try to satisfy request immediately
                                ;
                                ;  Now wait until message request is satisfied
                                ;
000514  113701  0000000         2$:    MOVB    CORUSR,R1       ;Get our job index number
000520  004737  0000000                 CALL    CHKABT          ;See if we have been aborted
000524  010105                         MOV     R1,R5           ;Get job index number
000526  062705  0000000                 ADD     #LCMQHD,R5      ;Point to list head of completed requests
000532  020415                 5$:     CMP     R4,(R5)         ;Is request block we are waiting for next?
000534  001413                         BEQ     3$              ;Br if yes
000536  005715                         TST     (R5)            ;Are we at the end of the list?
000540  001404                         BEQ     4$              ;Br if yes
000542  011505                         MOV     (R5),R5         ;Follow chain to next entry
```

```
      58 000544  062705  0000000                 ADD     #MR$LNK,R5      ;Point to cell with forward link
      59 000550  000770                          BR      5$              ;Continue following linked list
      60                                ;
      61                                ;  Request has not been satisfied yet.
      62                                ;  Suspend job until a message arrives.
      63                                ;
      64 000552  012700  0000000       4$:     MOV     #S$MSWT,R0      ;Put job in message-wait state
      65 000556  004737  0000000               CALL    QHDSPN          ;Change state and suspend job
      66                                ;
      67                                ;  Job has been restarted.
      68                                ;  See if request has been satisfied.
      69                                ;
      70 000562  000754                          BR      2$              ;Go check again to see if request satisfied
      71                                ;
      72                                ;  The request has been satisfied
      73                                ;  Remove the completed request block from the list.
      74                                ;
      75 000564  016415  0000000       3$:     MOV     MB$FLK(R4),(R5) ;Remove completed request block from list
      76                                ;
      77                                ;  Move the message text to the user's buffer
      78                                ;
      79 000570  010405                          MOV     R4,R5           ;Get address of request block
      80 000572  004737  001420'                 CALL    MSGMOV          ;Move message to user's buffer
      81 000576  010037  0000000                 MOV     R0,URO          ;Return message length to user in R0
      82                                ;
      83                                ;  Free the message request block and associated message text block
      84                                ;
      85 000602  004737  002352'                 CALL    FREMRB          ;Free message request and text blocks
      86                                ;
      87                                ;  Finished
      88                                ;
      89 000606  000207                  20$:    RETURN
```

```
                                        .SBTTL  MSRCPL -- Read message with completion routine
                                ;-----------------------------------------------------------------------
                                ;  This EMT is used to read a message with a completion routine signaling
                                ;  that a message has arrived.
                                ;
000610                          MSRCPL:
                                ;
                                ;  See if there is an existing message channel block for the specified
                                ;  message channel.
                                ;
000610  004737  001562'                CALL    MBSRCH          ;Search for existing message channel block
000614  103007                        BCC     1$              ;Br if found one
                                ;
                                ;  There is no existing message channel block.
                                ;  Create a new one.
                                ;
000616  004737  002152'                CALL    GETMCB          ;Get a new message channel block
000622  103004                        BCC     1$              ;Br if we got a message control block
000624  112737  000001  000000G       MOVB    #ERRO1,INTERR   ;Return error code 1
000632  000416                        BR      20$
                                ;
                                ;  Create a message request block for this job
                                ;
000634  004737  001710'         1$:    CALL    ADDMRB          ;Add message request block
000640  103006                        BCC     6$              ;Br if we got a message request block
                                ;
                                ;  Error, we could not get a free message request block
                                ;
000642  112737  000005  000000G       MOVB    #ERRO5,INTERR   ;Return error code 1
000650  004737  002272'                CALL    FREMCB          ;See if we need to free the message control bk
000654  000405                        BR      20$
                                ;
                                ;  Store address of user's completion routine in message request block
                                ;
000656  013764  000000C 000000G 6$:    MOV     EMTBLK+EB$RTN,MR$RTN(R4);Set address of user's compl routine
                                ;
                                ;  At this point we have created a message request block and added it to
                                ;  the list of pending request blocks for this channel.
                                ;  Call MSGACT to see if the request can be satisfied immediately.
                                ;   R2 = Address of message channel block.
                                ;   R4 = Address of message request block.
                                ;
000664  004737  000672'                CALL    MSGACT          ;Try to satisfy request immediately
                                ;
                                ;  Finished
                                ;
000670  000207                  20$:    RETURN
```

```
                                        .SBTTL  MSGACT -- Check message requests that may be satisfied
                                ;-----------------------------------------------------------------------
                                ;  MSGACT is called each time a message is queued for a message channel
                                ;  to see if there are any jobs waiting to receive a message from the
                                ;  channel.
                                ;  If there are waiting users, the users are activated.
                                ;
                                ;  Inputs:
                                ;   R2 = Pointer to message channel block.
                                ;
000672  010146                  MSGACT: MOV     R1,-(SP)
000674  010446                          MOV     R4,-(SP)
000676  010546                          MOV     R5,-(SP)
                                ;
                                ;  If there are no requesting users or no pending messages, return.
                                ;
000700  016205  0000000         2$:     MOV     MB$BUF(R2),R5   ;Are there any pending messages?
000704  001507                          BEQ     9$              ;Br if not
000706  016204  0000000                 MOV     MB$REQ(R2),R4   ;Are there any requesting users?
000712  001504                          BEQ     9$              ;Br if not
                                ;
                                ;  There are waiting users and pending messages.
                                ;  Remove the 1st message text block and the 1st request block from the
                                ;  lists for this message channel.
                                ;
000714  016562  0000000 0000000         MOV     MU$FLK(R5),MB$BUF(R2);Remove 1st pending message from list
000722  016462  0000000 0000000         MOV     MR$LNK(R4),MB$REQ(R2);Remove 1st request block
                                ;
                                ;  Assign the message to the request block
                                ;
000730  010564  0000000                 MOV     R5,MR$BUF(R4)   ;Set address of message text block in req blk
                                ;
                                ;  Add the request block to the list of pending request blocks for this job
                                ;
000734  116401  0000000                 MOVB    MR$JOB(R4),R1   ;Get # of job that is receiving message
000740  062701  0000000                 ADD     #LCMQHD,R1      ;Point to list head for job
000744  005711                  4$:     TST     (R1)            ;Have we reached end of list?
000746  001404                          BEQ     5$              ;Br if yes
000750  011101                          MOV     (R1),R1         ;Link to next pending request block
000752  062701  0000000                 ADD     #MR$LNK,R1      ;Point to cell with forward link
000756  000772                          BR      4$              ;Continue following chain
000760  010411                  5$:     MOV     R4,(R1)         ;Add new block to end of list
000762  005064  0000000                 CLR     MR$LNK(R4)      ;Say we are at the end of the chain
                                ;
                                ;  If the user wants a completion routine entered to signal message
                                ;  reception, queue the completion request.
                                ;
000766  026427  0000000 000001          CMP     MR$RTN(R4),#1   ;Is there a completion routine?
000774  101442                          BLOS    1$              ;Br if not
000776  004737  0000000                 CALL    GETQ            ;Get a completion queue entry (addr in R1)
001002  116400  0000000                 MOVB    MR$JOB(R4),R0   ;Get job index # of requesting job
001006  110061  0000000                 MOVB    R0,CQ$JOB(R1)   ;Set job # in completion queue entry
001012  116061  0000000 0000000         MOVB    LPRI(R0),CQ$PRI(R1);Set job priority
001020  112761  0000000 0000000         MOVB    #S$IOFN,CQ$RNS(R1) ;Set job execution state for cpl rtn
001026  112761  0000000 0000000         MOVB    #CP$STD,CQ$CP(R1);Set standard compl routine class priority
001034  005760  0000000                 TST     LITIME(R0)      ;Is this an interactive job?
001040  001403                          BEQ     3$              ;Br if not
```

```
     58 001042  112761  000000G 000000G          MOVB    #S$HICP,CQ$RNS(R1);Set interactive state
     59 001050  012761  001140' 000000G 3$:      MOV     #MSGCPL,CQ$RTN(R1);Set address of our completion routine
     60 001056  013761  000000G 000000G          MOV     @#KPAR5,CQ$PA5(R1);Set PAR 5 mapping value for this region
     61 001064  112761  000000G 000000G          MOVB    #QF$SCR,CQ$FLG(R1);Say compl routine is part of system
     62 001072  010104                          MOV     R1,R4           ;Get address of completion block for QCOMPL
     63 001074  004737  000000G                 CALL    QCOMPL          ;Queue completion request for waiting job
     64 001100  000677                          BR      2$              ;See if there are more requests to satisfy
     65                                 ;
     66                                 ;  There is no completion routine for waiting user.
     67                                 ;  Add completed wait block to pending list for user and
     68                                 ;  restart execution of waiting user.
     69                                 ;
     70 001102  116401  000000G         1$:     MOVB    MR$JOB(R4),R1   ;Get waiting user's job index number
     71 001106  026127  000000G 000000G         CMP     LSTATE(R1),#S$MSWT;Is job waiting for a message?
     72 001114  001271                          BNE     2$              ;Br if not
     73 001116  004737  000000G                 CALL    QHIPRI          ;Restart waiting user
     74 001122  000666                          BR      2$              ;See if more message requests to satisfy
     75                                 ;
     76                                 ;  There are no more message requests that can be satisfied.
     77                                 ;  If the message channel block is now idle, free it.
     78                                 ;
     79 001124  004737  002272'         9$:     CALL    FREMCB          ;Free message channel block if idle
     80                                 ;
     81                                 ;  Finished
     82                                 ;
     83 001130  012605                          MOV     (SP)+,R5
     84 001132  012604                          MOV     (SP)+,R4
     85 001134  012601                          MOV     (SP)+,R1
     86 001136  000207                          RETURN
```

```
                                        .SBTTL  MSGCPL -- Completion routine processing for messages
                                ;-------------------------------------------------------------------
                                ;  Whenever a message is received by a job that requested a message completion
                                ;  routine, two completion routines are triggered.
                                ;  The first completion routine executed is this routine (MSGCPL) which is
                                ;  the system completion routine that moves the message text to the user's
                                ;  buffer and then queues the user's completion routine.
                                ;
001140  010146                  MSGCPL: MOV     R1,-(SP)
001142  010246                          MOV     R2,-(SP)
001144  010446                          MOV     R4,-(SP)
001146  010546                          MOV     R5,-(SP)
                                ;
                                ;  Begin loop to process all completed message request blocks.
                                ;
001150  113702  0000000                 MOVB    CORUSR,R2       ;Get current job index number
001154  062702  0000000                 ADD     #LCMQHD,R2      ;Point to compl message request blk Q head
001160  011205                  1$:     MOV     (R2),R5         ;Get address of next completed request block
001162  001464                          BEQ     9$              ;Br if no more to process
001164  026527  0000000 000001          CMP     MR$RTN(R5),#1   ;Does this one have a compl routine?
001172  101004                          BHI     2$              ;Br if yes
001174  010502                          MOV     R5,R2           ;Link on and continue search
001176  062702  0000000                 ADD     #MR$LNK,R2
001202  000766                          BR      1$
                                ;
                                ;  We have found a completed message request block for this job.
                                ;  Remove the completed request block from the list for the job.
                                ;
001204  016512  0000000         2$:     MOV     MR$LNK(R5),(R2) ;Remove request block from list
                                ;
                                ;  Move the message to the user's buffer.
                                ;
001210  004737  001420'                 CALL    MSGMOV          ;Move message to user's buffer
001214  010004                          MOV     R0,R4           ;Save message length in R4
                                ;
                                ;  Queue a completion request to call the user's completion routine
                                ;
001216  004737  0000000                 CALL    GETQ            ;Get a completion queue entry (addr in R1)
001222  010461  0000000                 MOV     R4,CQ$R0(R1)    ;Pass message length to user in R0
001226  016500  0000000                 MOV     MR$BUF(R5),R0   ;Get address of message text buffer
001232  116000  0000000                 MOVB    MU$JOB(R0),R0   ;Get job index # of job that sent message
001236  006200                          ASR     R0              ;Convert to job number
001240  010061  0000000                 MOV     R0,CQ$R1(R1)    ;Pass sending job number in R1
001244  116500  0000000                 MOVB    MR$JOB(R5),R0   ;Get job index # of requesting job
001250  110061  0000000                 MOVB    R0,CQ$JOB(R1)   ;Set job # in completion queue entry
001254  116061  0000000 0000000        MOVB    LPRI(R0),CQ$PRI(R1);Set job priority
001262  112761  0000000 0000000        MOVB    #S$IOFN,CQ$RNS(R1) ;Set job execution state for cpl rtn
001270  112761  0000000 0000000        MOVB    #CP$STD,CQ$CP(R1);Set standard compl rtn class priority
001276  005760  0000000                 TST     LITIME(R0)      ;Is this an interactive job?
001302  001403                          BEQ     3$              ;Br if not
001304  112761  0000000 0000000        MOVB    #S$HICP,CQ$RNS(R1);Set interactive state
001312  016561  0000000 0000000 3$:     MOV     MR$RTN(R5),CQ$RTN(R1);Set address of completion routine
001320  010104                          MOV     R1,R4           ;Get address of completion block for QCOMPL
001322  004737  0000000                 CALL    QCOMPL          ;Queue completion request for waiting job
                                ;
                                ;  Now free the message request block and the associated message text block
                                ;
```

```
     58 001326  004737  002352'                    CALL    FREMRB          ;Free the message request block
     59                                ;
     60                                ;  Go back and see if there are any more completed message request
     61                                ;  blocks for this job.
     62                                ;
     63 001332  000712                          BR      1$
     64                                ;
     65                                ;  Finished all completed message request blocks for this job
     66                                ;
     67 001334  012605                  9$:     MOV     (SP)+,R5
     68 001336  012604                          MOV     (SP)+,R4
     69 001340  012602                          MOV     (SP)+,R2
     70 001342  012601                          MOV     (SP)+,R1
     71 001344  000207                          RETURN
```

```
                                        .SBTTL  MSGGET -- Move message text into internal buffer
                                ;-----------------------------------------------------------------
                                ;  MSGGET is called to move a message text string from the user's buffer
                                ;  into an internal buffer.
                                ;
                                ;  Inputs:
                                ;   R3 = Address of buffer in user's program space
                                ;   R4 = Address of internal buffer where message is to be stored
                                ;   R5 = Number of bytes to move
                                ;
001346  010346                  MSGGET: MOV     R3,-(SP)
001350  010446                          MOV     R4,-(SP)
001352  010546                          MOV     R5,-(SP)
                                ;
                                ;  If the user's buffer address is even, do a word-by-word move
                                ;
001354  032703  000001                  BIT     #1,R3           ;Is user's buffer on word boundary?
001360  001007                          BNE     3$              ;Br if not
001362  005205                          INC     R5              ;Bound # bytes up to word multiple
001364  000241                          CLC                     ;Clear for rotate
001366  006005                          ROR     R5              ;Get # words to move
001370  106523                  2$:     MFPD    (R3)+           ;Get a word from user's buffer
001372  012624                          MOV     (SP)+,(R4)+     ;Move to internal buffer
001374  077503                          SOB     R5,2$           ;Loop if more words to move
001376  000404                          BR      9$
                                ;
                                ;  Do byte-by-byte move
                                ;
001400  004737  0000000         3$:     CALL    GETUCH          ;Get byte from user's buffer
001404  110024                          MOVB    R0,(R4)+        ;Move to internal buffer
001406  077504                          SOB     R5,3$           ;Loop if more to move
                                ;
                                ;  Finished
                                ;
001410  012605                  9$:     MOV     (SP)+,R5
001412  012604                          MOV     (SP)+,R4
001414  012603                          MOV     (SP)+,R3
001416  000207                          RETURN
```

```
                                        .SBTTL  MBSRCH -- Search for message channel block
                                ;-----------------------------------------------------------------------
                                ;  MBSRCH is called to attempt to find an existing message channel
                                ;  block that matches the message channel whose name is specified
                                ;  by the current EMT.
                                ;
                                ;  Outputs:
                                ;   C-flag cleared ==> Message channel block found.
                                ;   C-flag set     ==> Could not find existing message channel block.
                                ;   R2 = Address of message channel block if one found.
                                ;
001562  010346                  MBSRCH: MOV     R3,-(SP)
001564  010446                          MOV     R4,-(SP)
                                ;
                                ;  Set up message channel name in CHNNAM
                                ;
001566  004737  001646'                 CALL    MSGNAM          ;Set up message channel name
                                ;
                                ;  Begin loop to search for existing message channel block
                                ;
001572  013702  000002'                 MOV     MBMHD,R2        ;Point to 1st message channel block
001576  001417                          BEQ     8$              ;Br if no message channel blocks
                                ;
                                ;  Compare message channel names
                                ;
001600  010203                  1$:     MOV     R2,R3           ;Get address of message channel block
001602  062703  0000000                 ADD     #MB$NAM,R3      ;Point to name in channel block
001606  012700  000012'                 MOV     #CHNNAM,R0      ;Point to target name
001612  012704  000006                  MOV     #6,R4           ;Names are 6 chars long
001616  122023                  2$:     CMPB    (R0)+,(R3)+     ;Compare names
001620  001003                          BNE     3$              ;Br if mismatch
001622  077403                          SOB     R4,2$           ;Loop to compare rest of names
                                ;
                                ;  We found the message channel block
                                ;
001624  000241                          CLC                     ;Signal success
001626  000404                          BR      9$
                                ;
                                ;  Names do not match.
                                ;  See if there are more blocks to check.
                                ;
001630  016202  0000000          3$:     MOV     MB$FLK(R2),R2   ;Get address of next block
001634  001361                          BNE     1$              ;Loop if there is another one
                                ;
                                ;  We cannot find the message channel block
                                ;
001636  000261                  8$:     SEC                     ;Signal failure on return
                                ;
                                ;  Finished
                                ;
001640  012604                  9$:     MOV     (SP)+,R4
001642  012603                          MOV     (SP)+,R3
001644  000207                          RETURN
```

```
                                        .SBTTL  MSGNAM -- Set up name of message channel
                                ;-----------------------------------------------------------------------
                                ;  MSGNAM is called to acquire the message channel name specified
                                ;  with the current EMT and move it to a temporary buffer.
                                ;
                                ;  Outputs:
                                ;   CHNNAM = Message channel name.
                                ;
001646  010246                  MSGNAM: MOV     R2,-(SP)
001650  010346                          MOV     R3,-(SP)
001652  010446                          MOV     R4,-(SP)
                                ;
                                ;  Move message channel name from user's area to internal temp cell
                                ;
001654  013703  000000C                 MOV     EMTBLK+EB$NAM,R3;Get pointer to user's name cell
001660  012702  000012'                 MOV     #CHNNAM,R2      ;Get pointer to temp cell
001664  012704  000006                  MOV     #6,R4           ;Move 6 chars
001670  004737  0000000         1$:     CALL    GETUCH          ;Get char from user's buffer
001674  110022                          MOVB    R0,(R2)+        ;Move to temp buffer
001676  077404                          SOB     R4,1$           ;Loop if more to move
                                ;
                                ;  Finished
                                ;
001700  012604                          MOV     (SP)+,R4
001702  012603                          MOV     (SP)+,R3
001704  012602                          MOV     (SP)+,R2
001706  000207                          RETURN
```

```
                                        .SBTTL  ADDMRB -- Add new message request block
                                ;-----------------------------------------------------------------
                                ;  ADDMRB creates a new message request block and adds it to the list
                                ;  of message request blocks for a specified message channel block.
                                ;  Information about the current job number, user's buffer address and
                                ;  buffer size are set up in the message request block.
                                ;
                                ;  Inputs:
                                ;   R2 = Address of message channel block.
                                ;
                                ;  Outputs:
                                ;   C-flag cleared ==> We successfully added a message request block.
                                ;   C-flag set     ==> Error detected while creating message request block.
                                ;   R4 = Address of message request block we created.
                                ;
001710                          ADDMRB:
                                ;
                                ;  Get a message request block off of the free list
                                ;
001710  013704  000006'                 MOV     MRBFHD,R4       ;Get message request block off of free list
001714  001436                          BEQ     8$              ;Br if there are no free blocks
001716  016437  000000G 000006'         MOV     MR$LNK(R4),MRBFHD;Remove block from free list
                                ;
                                ;  We got a free message request block.
                                ;  Initialize with values from EMT argument block.
                                ;
001724  013764  000000C 000000G         MOV     EMTBLK+EB$BUF,MR$UBA(R4) ;User's buffer address
001732  013764  000000C 000000G         MOV     EMTBLK+EB$SIZ,MR$UBS(R4) ;User's buffer size
001740  113764  000000G 000000G         MOVB    CORUSR,MR$JOB(R4) ;Job index number
001746  005064  000000G                 CLR     MR$RTN(R4)      ;Say no completion routine
001752  005064  000000G                 CLR     MR$BUF(R4)      ;Say no associated text block
                                ;
                                ;  Add message request block to tail of request blocks for message channel
                                ;
001756  010200                          MOV     R2,R0           ;Get address of message channel block
001760  062700  000000G                 ADD     #MB$REQ,R0      ;Point to list head of request blocks
001764  005710                  2$:     TST     (R0)            ;Are we at the end of the list?
001766  001404                          BEQ     3$              ;Br if yes
001770  011000                          MOV     (R0),R0         ;Link to next one on list
001772  062700  000000G                 ADD     #MR$LNK,R0      ;Point to cell with forward link
001776  000772                          BR      2$              ;Continue following chain
002000  010410                  3$:     MOV     R4,(R0)         ;Add new block to end of list
002002  005064  000000G                 CLR     MR$LNK(R4)      ;Say we are at end of list
002006  000241                          CLC                     ;Signal success on return
002010  000401                          BR      9$
                                ;
                                ;  Error -- No free message request blocks
                                ;
002012  000261                  8$:     SEC                     ;Signal failure on return
                                ;
                                ;  Finished
                                ;
002014  000207                  9$:     RETURN
```

```
                                        .SBTTL  MSGABT -- Abort all message requests for a job
                                ;-----------------------------------------------------------------------
                                ;  MSGABT is called when a job aborts to clean up all pending message
                                ;  requests for the job.
                                ;
                                ;  Inputs:
                                ;   R1 = Job index number of aborting job.
                                ;
002016  010246                  MSGABT: MOV     R2,-(SP)
002020  010346                          MOV     R3,-(SP)
002022  010446                          MOV     R4,-(SP)
002024  010546                          MOV     R5,-(SP)
                                ;
                                ;  Free all completed message request blocks that are pending for the job
                                ;
002026  016105  0000000                 MOV     LCMQHD(R1),R5   ;Get addr of 1st completed message request
002032  001410                          BEQ     2$              ;Br if there are none
002034  016504  0000000         1$:     MOV     MR$LNK(R5),R4   ;Get address of next request block
002040  004737  002352'                 CALL    FREMRB          ;Free the message request block
002044  010405                          MOV     R4,R5           ;Get address of next completed block
002046  001372                          BNE     1$              ;Loop if there is another block to free
002050  005061  0000000                 CLR     LCMQHD(R1)      ;Say there are no completed requests for job
                                ;
                                ;  Free all pending message requests for this job
                                ;
002054  013702  000002'         2$:     MOV     MBMHD,R2        ;Get addr of 1st message channel block
002060  001427                          BEQ     9$              ;Br if there are none
002062  010203                          MOV     R2,R3           ;Get address of message channel block
002064  062703  0000000         7$:     ADD     #MB$REQ,R3      ;Get address of cell with ptr to request blk
002070  011305                  6$:     MOV     (R3),R5         ;Get address of next message request block
002072  001414                          BEQ     4$              ;Br if there are no more
002074  120165  0000000                 CMPB    R1,MR$JOB(R5)   ;Is this message request for aborting job?
002100  001005                          BNE     5$              ;Br if not
002102  016513  0000000                 MOV     MR$LNK(R5),(R3) ;Remove message block from list
002106  004737  002352'                 CALL    FREMRB          ;Free the message request block
002112  000766                          BR      6$              ;Go see if there are more to free
002114  010503                  5$:     MOV     R5,R3           ;Get address of current block
002116  062703  0000000                 ADD     #MR$LNK,R3      ;Point to cell with forward link
002122  000762                          BR      6$              ;Keep looking for more requests for this job
002124  016203  0000000         4$:     MOV     MB$FLK(R2),R3   ;Get pointer to next message channel block
002130  004737  002272'                 CALL    FREMCB          ;See if we need to free current channel block
002134  010302                          MOV     R3,R2           ;Get address of next message channel block
002136  001352                          BNE     7$              ;Br if more message channels to check
                                ;
                                ;  Finished
                                ;
002140  012605                  9$:     MOV     (SP)+,R5
002142  012604                          MOV     (SP)+,R4
002144  012603                          MOV     (SP)+,R3
002146  012602                          MOV     (SP)+,R2
002150  000207                          RETURN
```

```
                                        .SBTTL  GETMCB -- Get a new message control block
                                ;-----------------------------------------------------------------
                                ;  GETMCB is called to get a new message control block.
                                ;
                                ;  Inputs:
                                ;   CHNNAM = Message channel name string
                                ;
                                ;  Outputs:
                                ;   C-flag cleared ==> Message control block gotten.
                                ;   C-flag set     ==> Could not get a free message control block.
                                ;   R2 = Address of message control block.
                                ;
002152                          GETMCB:
                                ;
                                ;  Get 1st message control block off of free list
                                ;
002152  013702  000004'                 MOV     MCBFHD,R2       ;Get 1st free message control block
002156  001427                          BEQ     8$              ;Br if there are no free message blocks
002160  016237  000000G 000004'         MOV     MB$FLK(R2),MCBFHD ;Remove block from free list
                                ;
                                ;  We got a free message control block.
                                ;  Initialize it.
                                ;
002166  005062  000000G                 CLR     MB$BUF(R2)      ;No pending messages
002172  005062  000000G                 CLR     MB$REQ(R2)      ;No pending message requests
002176  013762  000012' 000000G         MOV     CHNNAM,MB$NAM(R2);Set 1st 2 chars of name
002204  013762  000014' 000002G         MOV     CHNNAM+2,MB$NAM+2(R2);Set 2nd 2 chars of name
002212  013762  000016' 000004G         MOV     CHNNAM+4,MB$NAM+4(R2);Set 3rd 2 chars of name
                                ;
                                ;  Add message channel block to list of active message channels
                                ;
002220  013762  000002' 000000G         MOV     MBMHD,MB$FLK(R2);Add new block to list of active blocks
002226  010237  000002'                 MOV     R2,MBMHD        ;Put new block at head of list
                                ;
                                ;  We successfully got a block
                                ;
002232  000241                          CLC                     ;Signal success on return
002234  000401                          BR      9$
                                ;
                                ;  There are no free message control blocks
                                ;
002236  000261                  8$:     SEC                     ;Signal failure on return
                                ;
                                ;  Finished
                                ;
002240  000207                  9$:     RETURN
```

```
                                        .SBTTL  GETMTB -- Get a free message text block
                                ;-----------------------------------------------------------------------
                                ;  GETMTB is called to try to acquire a free message text block.
                                ;  Message text blocks are used to hold message text strings.
                                ;
                                ;  Outputs:
                                ;   C-flag cleared ==> A free message text block was gotten.
                                ;   C-flag set     ==> There are no free message text blocks.
                                ;   R4 = Address of message text block if one was gotten.
                                ;
002242                          GETMTB:
                                ;
                                ;  Get 1st message text block off of free list
                                ;
002242  013704  000010'                 MOV     MTBFHD,R4       ;Get pointer to 1st free message text block
002246  001407                          BEQ     1$              ;Br if there are no free blocks
002250  016437  0000000 000010'         MOV     MU$FLK(R4),MTBFHD ;Remove block from free list
002256  005064  0000000                 CLR     MU$FLK(R4)      ;Say block is not part of any list
002262  000241                          CLC                     ;Signal success on return
002264  000401                          BR      9$
                                ;
                                ;  There are no free message text blocks
                                ;
002266  000261                 1$:     SEC                     ;Signal failure on return
                                ;
                                ;  Finished
                                ;
002270  000207                 9$:     RETURN
```

```
                                        .SBTTL  FREMCB -- Free a message control block
                                ;-----------------------------------------------------------------------
                                ;  FREMCB is called to attempt to return a message control block to the
                                ;  free list.  This routine checks to see if there are any pending
                                ;  messages for the message channel or pending message requests.
                                ;  If there are any pending messages or message requests, this routine
                                ;  returns without freeing the message control block.
                                ;
                                ;  Inputs:
                                ;   R2 = Address of message control block to be freed.
                                ;
002272                          FREMCB:
                                ;
                                ;  See if this message control block is idle
                                ;
002272  005762  000000                  TST     MB$BUF(R2)      ;Any pending messages?
002276  001024                          BNE     9$              ;Br if yes
002300  005762  000000                  TST     MB$REQ(R2)      ;Any pending message requests?
002304  001021                          BNE     9$              ;Br if yes
                                ;
                                ;  This message control block is idle.
                                ;  Remove it from the active list.
                                ;
002306  012700  000000C                 MOV     #MBMHD-MB$FLK,R0;Point to active list head
002312  020260  000000    1$:           CMP     R2,MB$FLK(R0)   ;Are we next block on list?
002316  001404                          BEQ     2$              ;Br if yes
002320  016000  000000                  MOV     MB$FLK(R0),R0   ;Chain to next block
002324  001372                          BNE     1$              ;Go check it
002326  000403                          BR      3$              ;Block is not on active list
002330  016260  000000  000000  2$:     MOV     MB$FLK(R2),MB$FLK(R0);Remove from active list
                                ;
                                ;  Return it to free list.
                                ;
002336  013762  000004' 000000  3$:     MOV     MCBFHD,MB$FLK(R2);Add block to free list
002344  010237  000004'                 MOV     R2,MCBFHD
                                ;
                                ;  Finished
                                ;
002350  000207                  9$:     RETURN
```

```
                                        .SBTTL  FREMRB -- Free a message request block
                                ;-----------------------------------------------------------------
                                ;  FREMRB is called to free a message request block.
                                ;  It also frees any associated message text block.
                                ;
                                ;  Inputs:
                                ;   R5 = Address of message request block being freed.
                                ;
002352                          FREMRB:
                                ;
                                ;  See if there is an associated message text block that needs to be freed
                                ;
002352  016500  000000G               MOV     MR$BUF(R5),R0   ;Is there an associated message text block?
002356  001407                       BEQ     1$              ;Br if not
002360  013760  000010' 000000G      MOV     MTBFHD,MU$FLK(R0);Add message text block to free list
002366  010037  000010'               MOV     R0,MTBFHD
002372  005065  000000G              CLR     MR$BUF(R5)      ;Say message text block has been freed
                                ;
                                ;  Free the message request block
                                ;
002376  013765  000006' 000000G 1$:  MOV     MRBFHD,MR$LNK(R5);Add block to free list
002404  010537  000006'              MOV     R5,MRBFHD
                                ;
                                ;  Finished
                                ;
002410  000207                       RETURN
                                ;
                                ;  Top of TSMSG
                                ;
002412                          MSGTOP:
000001                                  .END
Errors detected:  0

*** Assembler statistics


Work  file  reads:  0
Work  file writes:  0
Size of work file:  64 Words  ( 1 Pages)
Size of core pool:  1720 Words  ( 7 Pages)
Operating  system:  RT-11

Elapsed time:  00:00:13.40
DK:TSMSG,LP:TSMSG=DK:TSMSG.MAC/C/N:SYM
```

Cross reference table (CREF V05.04)

```
ADDMRB    5-30      6-24      13-16#
CHKABT    1-28      5-50
CHNNAM    1-48#    11-28      12-16     15-26     15-27     15-28
CORUSR    1-25      3-49       5-49      8-16     13-29
CP$STD    1-32      7-55       8-48
CQ$CP     1-32      7-55*      8-48*
CQ$FLG    1-29      7-61*
CQ$JOB    1-31      7-52*      8-45*
CQ$PA5    1-32      7-60*
CQ$PRI    1-27      7-53*      8-46*
CQ$RO     1-28      8-39*
CQ$R1     1-32      8-43*
CQ$RNS    1-31      7-54*      7-58*     8-47*     8-51*
CQ$RTN    1-29      7-59*      8-52*
EB$BUF    1-24      3-61      13-27
EB$NAM    1-26     12-15
EB$RTN    1-24      6-35
EB$SIZ    1-28      3-53      13-28
EMTBLK    1-24      3-53       3-61      5-11      6-35     12-15     13-27     13-28
ERR01     1-36#     3-21       5-25      6-19
ERR02     1-37#     3-29
ERR03     1-38#     4-23
ERR04     1-39#     3-56      10-29
ERR05     1-40#     5-35       6-29
FREMCB    3-30      5-36       6-30      7-79     14-41     17-12#
FREMRB    5-85      8-58      14-19     14-35     18-9#
GETMCB    3-19      5-23       6-17     15-13#
GETMTB    3-27     16-11#
GETQ      1-27      7-50       8-38
GETUCH    1-26      9-29      12-18
INTERR    1-28      3-21*      3-29*     3-56*     4-23*     5-25*     5-35*     6-19*     6-29*     10-29*
KPAR5     1-32      7-60
LCMQHD    1-31      5-52       7-36      8-17     14-16     14-22*
LITIME    1-27      7-56       8-49
LPRI      1-30      1-30       7-53      8-46
LSTATE    1-31      7-71
MB$$SZ    1-28      2-22       2-26*
MB$BUF    1-24      3-39       4-17      7-17      7-26*    15-24*    17-16
MB$FLK    1-24      1-25       2-23*     2-26*     5-75     11-42     14-40     15-19     15-32*    17-24     17-25     17-27
          17-30     17-30*     17-34*
MB$NAM    1-26     11-27      15-26*    15-27*    15-28*
MB$REQ    1-27      7-19       7-27*    13-36     14-29     15-25*    17-18
MBMHD     1-20      1-44#     11-21     14-26     15-32     15-33*    17-24
MBSRCH    3-13      4-11       5-17      6-11     11-12#
MCBFHD    1-45#     2-19*     15-17     15-19*    17-34     17-35*
MR$$SZ    1-30      2-34       2-38*
MR$BUF    1-30      7-31*      8-40     10-21     13-31*    18-13     18-17*
MR$JOB    1-25      7-35       7-51      7-70      8-44     13-29*    14-32
MR$LNK    1-25      2-35*      2-38*     5-58      7-27      7-40      7-43*     8-23      8-29      13-22     13-40     13-43*
          14-18     14-34      14-38     18-21*
MR$RTN    1-25      6-35*      7-48      8-20      8-52     13-30*
MR$UBA    1-31     10-24      13-27*
MR$UBS    1-31     10-25      13-28*
MRBFHD    1-46#     2-32*     13-20     13-22*    18-21     18-22*
MSEND     1-19      3-8#
MSGABT    1-20     14-9#
```

Cross reference table (CREF V05.04)

```
MSGACT    3-70      5-45      6-43      7-11#
MSGCK     1-19      4-7#
MSGCPL    7-59      8-9#
MSGGET    3-62      9-11#
MSGINI    1-20      2-8#
MSGMOV    5-80      8-33     10-12#
MSGNAM   11-17     12-9#
MSGTOP    1-20      2-15     18-30#
MSGWT     1-19      4-18      5-6#
MSRCPL    5-12      6-6#
MTBFHD    1-47#     2-45*    16-15     16-17*    18-15     18-16*
MU$FLK    1-31      2-57*     2-61*     3-43      7-26     16-17     16-18*    18-15*
MU$JOB    1-32      3-49*     8-41
MU$SIZ    1-27      3-58*    10-22
MU$TXT    1-29      2-47      3-60     10-23
PUTUCH    1-26     10-55
QCOMPL    1-27      7-63      8-54
QF$SCR    1-25      7-61
QHDSPN    1-28      5-65
QHIPRI    1-29      7-73
S$HICP    1-26      7-58      8-51
S$IOFN    1-26      7-54      8-47
S$MSWT    1-30      5-64      7-71
TSMSG     1-15#     1-19
URO       1-29      4-24*     5-81*
VMAXMC    1-24      2-20
VMSCHR    1-29      2-46      3-54      3-57
VMXMRB    1-30      2-30
VMXMSG    1-30      2-42      2-44*     2-56*
VPAR6     1-29      2-49
```